2005 NASA SOFTWARE OF THE YEAR SUMMARY EVALUATION DOCUMENT FORM
Instructions

Identification Information	
Software Title:	MACS: Multi Aircraft Control System (supplemented with ADRS: Aeronautical Datalink and Radar Simulator)
NASA Case No.	ARC-14776/MACS
Responsible Center(s):	Ames Research Center

Software's Developmental Status	
Current Technology Readiness Level (1-9): 8	Classification (A-H): E

Significance to NASA Mission Part A - Impact on NASA's Mission

The Multi Aircraft Control System (MACS) and the Aeronautical Datalink and Radar Simulator (ADRS) consitute an original mixed-fidelity simulation environment for rapid prototyping and evaluation of envisioned air traffic operational concepts. It contributes daily and directly to all four key objectives of the Aeronautics Research Mission Directorate:

1. Re-establish our commitment to mastering the science of subsonic (rotary and fixed wing), supersonic, and hypersonic flight;

The MACS/ADRS software provides a long-term platform for cutting-edge scientific research on the interaction of all air vehicles with other components of the air transportation system. The success and continued extensive use of this capability at NASA and its distribution to other government, industry and educational institutions represents a firm commitment and demonstrates NASA's unique capabilities to master the science of flight.

2. Preserve the Agency's research facilities, such as wind tunnels, as national assets

The software is the primary simulation platform at several NASA research laboratories; It is also used to supplement operator stations and integrate existing research facilities like full mission flight simulators at NASA Ames and Langley Research Centers, the Future Flight Central tower facility, and the CTAS laboratories with high fidelity air traffic control facilities and flight deck research laboratories. With laboratories equipped with this world class software and cross-facility integration NASA facilities are the best air transportation research environment in the world. This is demonstrated for example by: The FAA chose NASA laboratories that use MACS and ADRS software for air traffic operations research. Prestigious conferences gave best paper awards for research that used this novel simulation environment as the primary platform and integrated other NASA facilities via the ADRS networking infrastructure.

3. Focus research in areas that are appropriate to NASA's unique capabilities;

With MACS and ADRS software as a world-class simulation platform that hosts many advanced automation features and comprehensive data collection capabilities researchers can focus on developing and evaluating visionary concepts for the long term. The novel approach realized in the software enables experienced NASA scientists to quickly get a first glance at new concepts on their own office desktop. After refinements these concepts can be evaluated using the same software in a realistic meaningful multi-facility environment that only NASA can provide. MACS and ADRS provide all the tools necessary to conduct far-term research and already incorporate many of the functions envisioned for a 2025 environment. The software is uniquely designed to provide the look and feel of existing and envisioned systems and to determine specifications for operational systems, but not to be operationally used in the actual air traffic environment. Therefore research at NASA can progress at a much faster pace than at most other organizations and focus on the scientifically most challenging long term issues rather than immediate implementation problems that are more suitable for industry to resolve.

4. Directly address the needs of the Next Generation Air Transportation System (NGATS) in partnership with the Federal Aviation Administration (FAA) and other government agencies.

The MACS/ADRS software is designed to directly address the pressing research needs associated with NGATS operations like no other simulation software in the world. NGATS operations can be rapidly prototyped with MACS and ADRS. The software repertoire includes numerous NGATS key elements, such as 4D trajectory-based operations, equivalent visual operations, and super-dense operations. Furthermore, it already incorporates prototypes for automatic separation assurance. Results of the research conducted on Distributed Air/Ground Traffic Management using the software have shaped development of the NGATS vision. The software is designed to be shared among partners. This simplifies coordination and cooperation between government agencies and industry. Simulations can be conducted across multiple agencies, and the results and the tool prototypes can easily be distributed.
Significance to Science, Technology, & Industry in General Part B – Impact on Science & Technology
MACS and ADRS were developed in four years by a few people to fill a huge gap that has plagued air traffic operations research over several decades: The ability to rapidly prototype and simulate air/ground operations in the NAS comprehensively and produce meaningful research results on low cost platforms. The software has already produced results that have been worldwide recognized as significant advancements in the state of the art in aviation research and simulation technologies. It has also had a primary impact on education. The software was instrumental in creating a novel approach for teaching middle school students math problems. It is used to create a new Master's program at California State University Long Beach. As a byproduct of meeting the research demands of future concepts, accurate emulations of current day air traffic controller workstations and flight deck capabilities were created in MACS. The FAA is currently assessing to use this capability for cost-effective training of the large number of air traffic controllers that need to be hired over the next years.
Significance in Impact on the Quality of Human Life Part C
The FAA predicts one billion passengers by the year 2015. In 2005 the number of air passengers was a record 739 million, up from 690 million the previous year. Therefore the quality of life for almost everybody will suffer significantly, if the capacity necessary to deal with this increase cannot be achieved in time. The MACS/ADRS system is uniquely capable of simulating air traffic operations of today and the future by implementing alternative solutions and examining the effectiveness in a full mission environment. This will identify benefits and problems early. The effectiveness of this approach was demonstrated during the Advanced Air Transportation Technologies Program when researchers at ARC using MACS and ADRS were able to evaluate three different future Distributed Air/Ground Traffic Management concept elements in full mission air/ground simulations within one year. One of these experiments was a joint effort with NASA LaRC and represents the most ambitious and comprehensive air/ground experiment to date. During the successful experiment up to 38 MACS operator stations provided the entire ATC simulation, 70 % of the air traffic, the overall experiment control and data analysis stations. Eight airline pilots flew MACS flight decks equipped with Cockpit Displays of Traffic Information (CDTI). A network of 9 ADRS processes provided the data communication, and the data link and surveillance simulation for all components The air traffic operations laboratory (ATOL) at LaRC with 12 flight deck stations and additional background traffic and a full mission simulator at NASA ARC were also integrated via the ADRS. During the experiment almost three times current day traffic levels

were simulated with a new distribution of roles and responsibilities and highly sophisticated automation on the ground and on-board flight decks. According to forecasts and the Department of Transportation tripling capacity will be required in the next 20 years. This system has proven capable to conduct the research today. This research is instrumental in validating that controllers and pilots will be able to operate a future system that is safe, efficient, and environmentally friendly for the benefit of the flying and non-flying public.
POC: Parimal Kopardekar, NASA Ames Research Center
POC: Richard Barhydt, NASA Langley Research Center
The software is used in several projects that focus on environmental impact, like reducing aircraft noise and emissions. Researchers at LaRC used the software for controller and pilot in the loop evaluations of new Continuous Descent Approaches within the Quiet Aircraft Technologies program. The ADRS was used for the data connection between the full mission flight simulator at LaRC and air traffic provided by the Langley FMS-Autoflight Simulation Tools for Windows (FASTWIN) program. MACS was used as air traffic controller display. The use of MACS and ADRS resulted in significant cost savings and will continue in the future.
POC: David H. Williams, NASA Langley Research Center
For the US Tailored Arrivals Initiative, a joint NASA/Boeing/United Airlines project with support form the FAA and the San Francisco noise office, an accurate simulation of oceanic flights landing in SFO with the impacted airspace was created in MACS within six months. This capability includes an impressive, fully functional emulation of the new oceanic air traffic control system called Advanced Technologies & Oceanic Procedures (ATOP) and was ready for tests before the actual system went into operational use at Oakland. The simulation of Tailored Arrivals operations is used to ensure that new fuel efficient low noise/emissions arrivals into the Bay Area will be operationally acceptable.
POC: Rob Mead, Boeing Phantom Works
POC: Nancy Smith, Richard Coppenbarger, NASA Ames Research Center
The FAA and Eurocontrol are assessing the benefits of changing the organizational structure of air traffic control facilities. To assess the pros and cons of using automation and a new multi sector position (MSP) instead of multiple radar associates the FAA and Principal Investigator Dr. Kevin Corker conducted an experiment comparing two MSP organizations with the traditional facility organizations at NASA ARC using the MACS and ADRS software. Both MSP alternatives and the required automation were rapidly prototyped in a few months and simulations with challenging traffic and weather problems were conducted observed by representatives from the FAA, Eurocontrol, MITRE and other organizations. All observers commented on the high quality of the research and the results will have an impact on the future air traffic control facility organization and ground automation in Europe and the United States.
POC: Kevin Corker, San Jose State University
The California State University Long Beach has created an air traffic control simulation laboratory based on MACS and ADRS software. The software has also stimulated the creation of a new, interdisciplinary Master of Science Human Factors degree at CSULB. Students have already started taking this opportunity and other universities and colleges have already requested the software for similar applications. This will educate a new generation of young professionals in Human Factors and Air Transportation and feed into the next generation of scientists.
POC: Tom Strybel, California State University Long Beach (562 985 5035)

Extent of Current and Potential Use

Current Use: The following is a list of *independent* organizations currently using MACS and ADRS. Non-Disclosure Agreements (NDA) or Space Act Agreements with full contact information are on file for each of the following:

Present Use Government:

1. NASA (ARC, LaRC, NTX) 2. FAA (Technical Center)

Present Use Non-Government:

Educational Institutions

3. San Jose State University 4. California State University Long Beach 5. Dowling College 6. California State University, Northridge 7. ASA Ames University Affiliated Research Center

Commercial Users

8. The Boeing Company (Boeing Phantom Works and Boeing Commercial Airplanes) 9. Titan Inc, 10. Spectrum Software 11. Sensis Corporation (Seagull Inc.) 12. Northop Grunman IT

Potential Use Government:

MACS and ADRS are currently in daily use by a number of NASA projects. The pressing NGATS research demands will increase the use of the software to even more projects and facilities inside and outside NASA. The FAA Technical Center has requested the software. The FAA directors of safety and enroute operations have initiated the assessment of MACS and ADRS as a training system for on the job training in air traffic control facilities and an on-site assessment probably Oakland Air Route Control Center will begin in a few months. Potentially, the software will be used at most air traffic facilities in the NAS.

Potential Use Non Government:

The use of the software as a low cost/high fidelity and extremely capable simulation environment is attractive to many institutions in industry or the educational field. It can be expected that many Universities and Colleges with aviation interest will request the software to educate students and conduct meaningful air traffic research. Additionally, more commercial users in the aviation field will benefit significantly from cooperating with NASA to turn prototypes into reality. Boeing as major aircraft manufacturer is using MACS and ADRS and Lockheed Martin as major air traffic system manufacturer has already initiated discussions and is expected to use the software in support of building the En Route Automation System ERAM, which will be the primary platform for NGATS technologies. With key aircraft and ground system manufacturers and the FAA using the NASA technologies MACS and ADRS, many more are expected to follow and a homogeneous development environment can be created.

An internet version of MACS is already available with user/password access control at **http://human-factors.arc.nasa.gov/ihi/research_groups/air-ground-integration/MacsWeb/HF/MacsWeb.html**. Acess information can be obtained from Dr. Thomas Prevot (tprevot@mail.arc.nasa.gov). If this version were publicly available practically any air traffic interested person in the world could become a potential user and participate in distributed simulations.

Usability of the Software

The usability assessment is twofold. 1. Usability of an installation by study participants, experimenters, and other operators. During a simulation session, air traffic control specialists, pilots and experimenters use different workstations running the software. Each station is easily started in a few seconds via single command shortcuts that specify the configuration files and operator modes to be used. During experimental runs, air traffic controllers and pilots continuously provide high usability ratings of the software. Experimenters can customize data collection settings, graphically edit traffic scenarios and monitor the health status.

Usability of the Software (continued)
2. Installation and maintenance at different sites. The internet version provides direct password protected access of the executable from any state of the art PC with an internet connection and JAVA installed. For complete installations typically CDs are delivered with read me files. A single installation takes only a few minutes and can easily be done without support. A configuration with multiple stations sometimes requires some support which is available via telephone or email. Once properly installed, the software typically does not require additional support from NASA, as it is very reliable and highly configurable. All setup options are selectable through graphic interfaces usually with help tooltips. A MACS assistant is available on the internet, describing a number of the functions. In very rare cases a bug is reported via email that can usually easily be fixed. Since the MACS executable is only about four mega bytes in size a new executable can be emailed easily and replace the old one. Software developers and experimenters can provide assistance on demand, which is rarely required.
Quality Factors Considered in Software
The MACS/ADRS system has a remarkable track record. Since the full completion and integration of all primary components in 2004 all scheduled data collection runs with the system were carried out as planned, the vast majority exactly at the planned start time. In more than 100 data collection runs, each typically in excess of one hour, with at least ten pilot and controller operators no failure was encountered that caused a termination of the data collection run after the traffic had been started. Furthermore, unlike earlier systems controllers do not need to be trained on basic workstation functions and confederate pilots can handle at least twice as many aircraft as before after a short one to two hour on the job training period. Study participants and observers are constantly impressed with the well integrated and highly responsive toolset and the high fidelity of the simulation environment. These very positive effects on efficiency and cost effectiveness of the research are the result of a deliberately chosen design. The ADRS is implemented in C and can be compiled and run on Solaris, Windows, LINUX and Mac OS platforms. Communication speed and reliability was paramount for this application and the system architecture is straightforward and rarely needs changes. MACS uses an object-oriented system architecture and the JAVA programming language to achieve complete platform independence, a cost-free programming and runtime environment, a state of the art language that enables very efficient software development, and an architecture and development platform that is future-oriented and fun for programmers to work with. One executable jar file is used for all installations in all platforms including the web-version. In the system design particular emphasis was put on reliability, error tolerance and failure recovery. The system is designed to easily deal with individual system or subsystem failures. Internal threads are restarted automatically if errors are detected. If an operator station fails it can be restarted without interrupting the rest of the simulation. Additionally, performance and responsiveness are paramount in the MACS design. Any operator station intended for real-time use, such as controller and flight deck stations, is optimized through a sophisticated thread management process for maximum performance and responsiveness. Since the stations are designed to analyze human factors, high fidelity user interfaces with innovative algorithms for best responsiveness are used throughout the system. The software is optimized to provide the correct look and feel and “visible system behavior” to operators. The MACS implementation does not reflect the implementation used in fielded systems. Instead more efficient innovative algorithms and methods were implemented

that have similar characteristics as fielded systems, but enable adding new advanced functions easily. This is made possible because the system is intended exclusively for use in simulation and not for direct use in fielded air traffic systems.	
Efforts to Transfer/Commercialize Software	
Description of Plan/Strategy to Transfer/Commercialize Software	The primary use will be for NGATS research in government agencies, NASA, FAA and to distribute it to industry to form efficient partnerships. Distribution to universities to foster education in the aeronautics domain. Provide the software to the FAA to reduce air traffic controller training cost and improve effectiveness. Probably license maintenance and adaptation of training system to industry
NASA Intellectual Property Status/Potential	NASA owns all rights to the software. Several innovative methods for distributed simulations can be patented. Individual prototypes of future air/ground functions can also be patented
Commercialization Potential for the software.	The software can be commercialized as low cost/high end training simulator and air traffic research capability. Less capable COTS systems cost at least 500,000 USD for one low cost 10 operator training installation. The entire research capability is unique and developing it somewhere else from scratch would probably cost 10 million+.
Dates Software released for commercial or program use	Software has not been released for commercial use
List all existing licenses and/or partnership agreements for the software	
Innovation (Creative New Features, Solutions, and Achievements)	
The simulation and rapid prototyping capabilities established by MACS and ADRS are truly groundbreaking. COTS software exists for flight simulators and air traffic control simulators that mimic portions of current day capabilities. There is no integrated research simulator for realistic air/ground operations besides this one. MACS/ADRS were designed and implemented from scratch because no existing capability appeared appropriate for human-in-the-loop research. MACS contains numerous original capabilities: it is the only fully portable (100% JAVA) air traffic simulator that is adequate for airline pilots and air traffic controllers, it incorporates the only existing ATOP (oceanic) controller station emulation, it includes the only accurate, but fully configurable and easily modifyable (DSR) Center Controller workstation, it is the only aircraft target generator with full flight management, required time of arrival and merging and spacing capabilities and realistic data link interfaces for hundreds of aircraft in the scenario. It is also the only system in the world that has a highly responsive route planning/conflict probing capability which is proven usable in high workload situations. It is the only prototype for easy to use trajectory negotiation capabilities integrated directly into the NAS radar position. MACS can demonstrate and evaluate the Advanced Airspace Concept conflict resolution logic and deal with self-separating aircraft in mixed environments. The full suite of capabilities runs with the same executable locally on a single PC, as a standalone web version, in small scale simulation or in large scale distributed simulations connected to other high fidelity systems. Prototyping new capabilities in MACS is the fastest, most cost-effective way to evaluate visionary ideas in a meaningful context. Many envisioned NGATS functions are already prototyped in MACS.	

Instructions

The purpose of the Summary Evaluation Document is to provide the Software of the Year (SOY) Panel Members with most of the information necessary to evaluate each nominated software package.

Each Center must submit a Summary Evaluation Document for each software package they nominate. The information provided in the attachment must:

- Fit on six printed pages. A page is a standard 8.5 x 11-inch piece of paper printed in 12 pitch Times Roman font with one-inch margins (top, bottom, and sides). **Note: The SOY Panel Members will only be given the first 6 pages of the Summary Evaluation Document submitted for each software package nominated.**
- Contain all sections of the Summary Evaluation Document form (the evaluation sheet used maps directly to the sections in the Summary Evaluation Document form).
- Be sufficiently focused and accurate to allow the SOY Panel Members to easily understand and score the nominated software. Please use the Glossary for an explanation of terms used in these guidelines and in the evaluation sheet.

There are eight sections on the evaluation sheet and eight corresponding sections in the Summary Evaluation Document form as follows:

Section	Title	Required Information
1. Refer to the glossary in Appendix I for a definition of terms used. 2. For Sections III, IV, V, VI, and VII, use as much space as needed to describe the areas in the Summary Evaluation Document form, however, do not exceed the 6-page limit on the total Summary Evaluation Document form.		
I	Identification Information	Provide: • Software title, same as that used in Form 1329 (Space Act Award Application). • NASA case number assigned during the processing of the NASA Disclosure of Invention and New Technology (Including Software) Form 1679, and • Responsible Center(s) which includes the Center sponsoring the software nomination for SOY award and all other Centers involved in developing the software.
II	Software's Developmental Status	Provide the current Technology Readiness Level (as defined in Appendix II) of the software. If the level is 6 or less the software will be automatically excluded from SOY competition.
III Part A	NASA Mission Significance and Impact	Describe the significance and impact (see definitions of significance and impact in the SOY Glossary) the software has on NASA's mission. Identify: • NASA Headquarters programs, projects and

Section	Title	Required Information
1. Refer to the glossary in Appendix I for a definition of terms used. 2. For Sections III, IV, V, VI, and VII, use as much space as needed to describe the areas in the Summary Evaluation Document form, however, do not exceed the 6-page limit on the total Summary Evaluation Document form.		
		technologies that are being directly supported by this software. • Client group. • Why the software is significant in the technology areas? • The software's impact in these areas.
III Part B	Science, Technology, & Industry Significance and Impact	Describe the significance and impact the software has on science, technology, & industry beyond direct support to NASA's missions (e.g., biotechnology, medicine, education, etc.). This refers to the adaptation of NASA mission technologies to secondary technology application areas for clientele different than those originally intended. These technology areas are known as horizontal technologies (see glossary). Identify: • The sciences and/or technologies that are being directly supported by this software. • Client group. • Why the software is significant in the horizontal technology application areas. • The software's impact in these areas.
III Part C	Impact on the Quality of Human Life	Describe the significance and impact the software has on the quality of human life. Consider such things as: • Intellectual impact • Environmental impact • Energy conservation impact • Tool to help improve human understanding of life • Health and safety impact • Improvement in processes such as: administrative, technical, research, educational, etc.
IV	Extent of Current and Potential Use	Describe the extent to which the software is supporting or has potential to support government & private sector efforts. For present use identify: • Federal, state, and/or local governments using the software. • Non-government (private sector) organizations using the software. • Points of contact for each government and non-

Section	Title	Required Information
1. Refer to the glossary in Appendix I for a definition of terms used. 2. For Sections III, IV, V, VI, and VII, use as much space as needed to describe the areas in the Summary Evaluation Document form, however, do not exceed the 6-page limit on the total Summary Evaluation Document form.		
		government organization using the software, including name, address, and phone number. For potential use identify: • Federal, state, and/or local governments that may make use of the software. • Non-government (private sector) organizations that may make use of the software. • Where and how the software's sponsoring organization intends to try to expand the use of the software. For both current and potential use identify the level of use (modest, average, above average and excellent as defined in the glossary of these instructions).
	Creativity	Components used to evaluate software creativity on the software evaluation sheet are: – The usability of the software (approximately 10 % of the creativity score) – The quality of the software package (approximately 40% of the creativity score) – The efforts made to commercialize the software (approximately 10% of the creativity score), and – Innovation produced in the development of the software (approximately 30% of the creativity score).
V	Usability of the Software	Describe key factors, which make the software easy for the end user to use. Specifically address: • Ease of use features that help the end-user understand system displays, input requirements, and outputs. • Technical support provided for problem consultation, trouble-shooting, debugging, fixes, maintenance, and enhancements. • Documentation available including help functions. • Training available. Describe the courses to include media used (e.g., classroom, web, videos, etc.) target audience and schedule for the next 12 months.
VI	Quality Factors Considered	Provide the justification used for selecting each of the

<table>
<tr><th>Section</th><th>Title</th><th>Required Information</th></tr>
<tr><td colspan="3">1. Refer to the glossary in Appendix I for a definition of terms used.
2. For Sections III, IV, V, VI, and VII, use as much space as needed to describe the areas in the Summary Evaluation Document form, however, do not exceed the 6-page limit on the total Summary Evaluation Document form.</td></tr>
<tr><td></td><td>in Developing the Software</td><td>following:
• Architecture (e.g., Object oriented, functional decomposition, etc.)
• Programming language(s) used
• Operating environment (e.g., operating system(s), hardware platform(s), web interactive interface(s), etc.)
Furthermore, describe the quality factors that were addressed in developing the software and the tradeoffs made between each factor listed:
• Reliability
• Function
• Performance - to include a description of the performance objectives and technical performance measures that were used. Also indicate if the original performance objectives were achieved.
• Reuse
• Maintainability
See the glossary included in these instructions for definitions of each of the above terms.</td></tr>
<tr><td>VII</td><td>Efforts to Transfer / Commercialize Software</td><td>Identify efforts made to transfer or commercialize the software including:
• Plan/strategy to transfer or commercialize the software. This should include, but is not limited to, establishing licensable IP, marketing the software for commercial use and licensing, and creating NASA/industry partnerships.
• IP status and potential of the software, including efforts to establish rights in inventions, copyrights and trademarks that are licensable by NASA.
• Commercialization potential assessed, including the identification of key market factors, commercial needs, and the suitability of the software.
• Date(s) the software was released for commercial use in accordance with NPD/NPG 2210.
• List all existing IP licenses associated with the software in a commercial environment or NASA/industry partnership agreements for the development /commercialization of the software.</td></tr>
</table>

Instructions

Section	Title	Required Information
1. Refer to the glossary in Appendix I for a definition of terms used. 2. For Sections III, IV, V, VI, and VII, use as much space as needed to describe the areas in the Summary Evaluation Document form, however, do not exceed the 6-page limit on the total Summary Evaluation Document form.		
VIII	Innovation	Describe the extent of innovation (newness, originality, and/or uniqueness) involved in developing the software. Specifically address: • The extent to which the software is a redevelopment of COTS equivalent software available in the market. If COTS equivalent software exists, state why the COTS was not used and why the equivalent software was developed. • Improvement/non-trivial modification to the state of the art that was made in developing the software. • Any advances in the state-of-the-art achieved by the software. • Any ground-breaking/original software technologies such as new or novel methods, techniques, languages, processes, etc.

2006 NASA SOFTWARE OF THE YEAR SUMMARY EVALUATION DOCUMENT FORM
APPENDIX I
GLOSSARY

Advances the State-of-the-Art: Software that significantly improves or updates currently existing concepts, operating environments, development tools, languages or new processes.

Assessment of Use: An evaluation of the extent of present use of the software and of potential use/marketability of the software. Levels of use or potential use may be defined as follows:

- Modest: less than $1.0 million of useful value.
- Average: between $1.0 million and $10 million of useful value.
- Above Average: between $10 million and $100 million of useful value.
- Excellent: over $100 million of useful value.

Copyright: A government issued grant of exclusive right to an author for an original work that is fixed in a tangible medium of expression, such as software. This right includes the right to exclude others from copying, distributing, and from developing other software derived from the copyright protected software.

COTS (Commercial Off The Shelf) Equivalent SW Available on Market: Are there any software products on the market that are equivalent in functionality and capability to the nominated software product

Creativity: See innovation. Components used to evaluate software creativity on the software evaluation sheet are:

– The usability of the software (approximately 10 % of the creativity score)
– The quality of the software package (approximately 40% of the creativity score)
– The efforts made to commercialize the software (approximately 10% of the creativity score), and
– Innovation produced in the development of the software (approximately 30% of the creativity score).

Development Status: The current Technology Readiness Level (TRL) of the software package. If the software is rated between 1 and 6, it is automatically disqualified from further SOY competition. The definitions of the TRL levels are found in Appendix II.

Documentation Quality: The degree to which published operating procedures, system functional descriptions, and technical specifications are understandable and useful.

Ease of Use: The end user's perspective of how effortless the system is to interact with and understand. This includes several user related issues such as:

– User system interface (e.g., a graphical user interface (GUI)) and the mechanisms (menus, icons and buttons) by which the user exercises the system functions,
– User support provided, and
– Flexibility in changing the content and format of system outputs (reports, displays, and other output).

2006 NASA SOFTWARE OF THE YEAR SUMMARY EVALUATION DOCUMENT FORM
APPENDIX I
GLOSSARY

Efforts to Commercialize Software: Patent council determination that the software may be licensable, patents, copyrighted material, trade secrets, inventions, trademarks and other knowledge that is the basis for commercializing the software.

Function: How closely the system processes match the end user's requirements. Also, refers to verification of the software program with regard to its correctness in meeting the requirements/specifications.

Ground Breaking/Original: Software applications whose functionality never existed before. This item refers to the development of new software technologies such as new languages, methods, techniques and processes.

Government Potential Use: The likelihood that the currently operational NASA software may be utilized in support of other government agencies (federal, state, or local).

Government Present Use: The extent of current federal, state, and/or local government utilization of the currently operational NASA software.

Horizontal Technology: A Technology in one technology area of application that is adapted to a different area of application.

Impact: The effect of the software on the program, and/or project. Examples of impact include: cost and timesavings, increased productivity, reduced risk, and increased security and safety,

Improvement/Non-Trivial Modification: New software or any pre-existing software modified by more than a trivial variation or improvement. A trivial variation or improvement includes minor code improvements that do not materially alter the software's operation.

Innovation: Producing meaningful new ideas, forms, methods, techniques, processes, systems, and interpretations or analogies. Also, using new knowledge, ideas, and/or inventions to create new products or services. Components used to evaluate software creativity on the software evaluation sheet are:
– Whether or not there is equivalent COTS software available,
– Improvement/non-trivial modification of previously existing software,
– Advances in the state-of-the-art, and
– Groundbreaking/original effort.

Invention: Any new idea, concept, technique, device, or process that has not yet been commercialized.

Justification for selecting technology and/or approach chosen: This justification is concerned with use of effective architecture(s), languages and tools. What efforts were made to select an architecture that would assure the optimal technological approach? For example:
– What was the architecture (Object-oriented, Function-based, etc) chosen and why?

– What language(s) (such as 4GLs or specialized languages) was chosen and why?

Maintainability: The ease and cost-effectiveness of system trouble-shooting, fixes, upgrades, and enhancements to meet changing system requirements.

NASA Case No: The number used in Form 1329 and is assigned by the Center Patent Attorney during processing of the New Technology Disclosure Form 1679.

Non-Government Potential Use: The likelihood that the currently operational NASA software may be utilized in the support of industry and non-profit sectors.

Non-Government Present Use: The extent of current utilization by industry and/or non-profit sectors of the currently operational NASA software.

Other Science and Technologies: Horizontal or crosscutting technology areas (e.g., Biotechnology, Communications, Construction, Education, Environment, Information Technology, Manufacturing, Materials, Medicine, etc) and secondary uses of the technology:

– Where the user(s) is not necessarily part of the clientele group for whom the application was originally developed.
– Whose application extends outside of NASA's mission support.

Patent: A government grant issued to an inventor or applicant for an invention that gives the inventor or applicant the right to exclude others from making, using, selling, or importing the patented invention.

Performance: The efficiency and effectiveness of the software system operation, in terms of responsiveness, throughput, cost and other technical performance measures. Response is a measure of how quickly and effectively the system reacts to a user's interaction with the system. Throughput is a measure of the computational work (based on workload characterization) accomplished by the system (software and hardware) within a specified time. The technical performance measures vary from system to system.

Portability: The extent of compatibility of the software with different operating system environments.

Quality: The extent of the superiority or excellence of the software measured by factors such as: how correctly the software performs the functions for which it was designed; system performance; system reliability; maintainability; and reuse of design, specifications and code.

Reliability: A measure of the probability that a system is operating satisfactorily at a given time. Also, refers to failsafe features built into the application.

Responsible Center: this is the sponsoring Center of the software nominated for the Software of the Year (SOY) Award.

APPENDIX I
GLOSSARY

Reuse: The extent to which the design, specifications, and/or source lines of certified software code of the system being considered for the SOY Award has been structured to facilitate adoption into systems to be developed in the future. Also, the extent to which previous designs, specifications, and/or source lines of certified software code have been incorporated into the system being considered for SOY award.

Science and Technology Significance: The extent of impact the software has on NASA's missions and/or the impact of the software on other science and Technology. See "Other Science and Technology" for further definition in this area.

Significance: Why something stands out or is important. Examples include: unique or greatly improved processes or products; functions, analytical tools and models that enable the development of systems or enable the execution of missions; and new and unique product that has a high probability of commercial success.

Software Class (from NPR 7150.2, NASA Software Engineering Requirements):

Class A Human Rated Software Systems	Applies to all space flight software subsystems (ground and flight) developed and/or operated by or for NASA to support human activity in space and that interact with NASA human space flight systems. Space flight system design and associated risks to humans are evaluated over the program's life cycle, including design, development, fabrication, processing, maintenance, launch, recovery, and final disposal. Examples of Class A software for human rated space flight include but are not limited to: guidance; navigation and control; life support systems; crew escape; automated rendezvous and docking; failure detection, isolation and recovery; and mission operations.
Class B Non-Human Space Rated Software Systems	Flight and ground software that must perform reliably in order to accomplish primary mission objectives. Examples of Class B software for non-human (robotic) spaceflight include, but are not limited to, propulsion systems; power systems; guidance navigation and control; fault protection; thermal systems; command and control ground systems; planetary surface operations; hazard prevention; primary

	instruments; or other subsystems that could cause the loss of science return from multiple instruments.
Class C Mission Support Software	Flight or ground software that is necessary for the science return from a single (non-critical) instrument or is used to analyze or process mission data or other software for which a defect could adversely impact attainment of some secondary mission objectives or cause operational problems for which potential work-arounds exist. Examples of Class C software include, but are not limited to, software that supports prelaunch integration and test, mission data processing and analysis, analysis software used in trend analysis and calibration of flight engineering parameters, primary/major science data collection and distribution systems, major Center facilities, data acquisition and control systems, aeronautic applications, or software employed by network operations and control (which is redundant with systems used at tracking complexes). Class C software must be developed carefully, but validation and verification effort is generally less intensive than for Class B.
Class D Analysis and Distribution Software	Non-space flight software. Software developed to perform science data collection, storage, and distribution; or perform engineering and hardware data analysis. A defect in Class D software may cause rework but has no direct impact on mission objectives or system safety. Examples of Class D software include, but are not limited to, software tools; analysis tools, and science data collection and distribution systems.
Class E Development Support Software	Non-space flight software. Software developed to explore a design concept; or support software or hardware development functions such as requirements management, design, test and integration, configuration management, documentation, or perform science analysis. A defect in Class E software may cause rework but

	has no direct impact on mission objectives or system safety. Examples of Class E software include, but are not limited to, earth science modeling, information only websites (non-business/information technology); science data analysis; and low technical readiness level research software.
Class F General Purpose Computing Software (Multi-Center or Multi-Program/Project)	General purpose computing software used in support of the Agency, multiple Centers, or multiple programs/projects, as described for the General Purpose Infrastructure To-Be Component of the NASA Architecture, Volume 5 (To-Be Architecture), and for the following portfolios: voice, wide area network, local area network, video, data centers, application services, messaging and collaboration, and public web. A defect in Class F software is likely to affect the productivity of multiple users across several geographic locations, and may possibly affect mission objectives or system safety. Mission objectives can be cost, schedule, or technical objectives for any work that the Agency performs. Examples of Class F software include, but are not limited to, software in support of the NASA-wide area network; the NASA Web portal; and applications supporting the Agency's Integrated Financial Management Program, such as the time and attendance system, Travel Manager, Business Warehouse, and E-Payroll.
Class G General Purpose Computing Software (Single Center or Project)	General purpose computing software used in support of a single Center or project, as described for locally deployed portions of the General Purpose Infrastructure To-Be Component of the NASA Architecture, Volume 5 (To-Be Architecture) and for the following portfolios: voice, local area network, video, data centers, application services, messaging and collaboration, and public web. A defect in Class G software is likely to affect the productivity of multiple users in a single geographic location or workgroup, but is unlikely to affect mission

	objectives or system safety. Examples of Class G software include, but are not limited to, software for Center custom applications such as Headquarters' Corrective Action Tracking System and Headquarters' ODIN New User Request System.
Class H: General Purpose Desktop Software	General purpose desktop software as described for the General Purpose Infrastructure To-Be Component (Desktop Hardware & Software Portfolio) of the NASA Architecture, Volume 5 (NASA To-Be Architecture). This class includes software for Wintel, Mac, and Unix desktops as well as laptops. A defect in Class H software may affect the productivity of a single user or small group of users but generally will not affect mission objectives or system safety. However, a defect in desktop IT-security related software, e.g., anti-virus software, may lead to loss of functionality and productivity across multiple users and systems. Examples of Class H software include, but are not limited to, desktop applications such as Microsoft Word, Excel, and Power Point, and Adobe Acrobat.

Technical Support: The support available for user assistance, trouble-shooting, fixes, upgrades, enhancements, and documentation.

Technology Commercialization: The process of new technology development through partnerships with government and industry with the objective of creating new products, processes, or services with commercial potential.

Technology Transfer: The process by which technology developed in one organization, in one area, or for one purpose is applied in another organization, in another area, or for another purpose

Technology Readiness Levels (TLR): The level of software system development. There are nine software technology readiness levels, ranging from 1 to 9, associated with the NASA software development life cycle and software having a TRL of 6 or less is automatically disqualified from the Software of the Year competition.

Software Title: the software title should be the same as that used in Form 1329 (Space Act Award Application).

APPENDIX I
GLOSSARY

Understandability: The degree to which the end-user can easily grasp the conceptual operation of the software (i.e., the system architecture). For example, can the end-user easily understand the system displays and outputs?

Usability: How well the user can apply the system functions to his/her needs. The software system usability attributes include understandability, ease-of-use, availability of technical support, quality end-user documentation, and availability of training.

2006 NASA SOFTWARE OF THE YEAR SUMMARY EVALUATION FORM
APPENDIX II
TECHNOLOGY READINESS LEVELS APPLIED TO SOFTWARE

TRL 9: Actual system "flight proven" through successful mission operations

Thoroughly debugged software. Fully integrated with operational hardware/software systems. All documentation has been completed and users have successful operational experience. Sustaining software-engineering support in place. Actual system fully demonstrated.

TRL 8: Actual system completed and "flight qualified" through test and demonstration (Ground or Flight)

Thoroughly debugged software. Fully integrated with operational hardware and software systems. Most user documentation, training documentation, and maintenance documentation completed. All functionality tested in simulated and operational scenarios. V&V completed.

TRL 7: System prototype demonstration in a relevant environment

Most of the software is functionality available for demonstration and test. Well integrated with operational hardware/software systems. Most software bugs removed. Limited documentation available.

TRL 6: System/subsystem model or prototype demonstration in a relevant environment (Ground or Space)

Prototype implementations if the software is on full-scale realistic problems. Partially integrated with existing hardware/software systems. Limited documentation available. Engineering feasibility fully demonstrated.

TRL 5: Component and/or breadboard validation in relevant environment

Prototype implementations. Experiments with realistic problems. Simulated interfaces to existing systems.

TRL 4: Component and/or breadboard validation in laboratory environment

Standalone prototype implementations. Experiments with full-scale problems or data sets.

TRL 3: Analytical and experimental critical function and/or characteristic proof-of-concept

Limited functionality implementations. Experiments with small representative data sets. Scientific feasibility fully demonstrated.

TRL 2: Technology concept and/or application formulated

Basic principles coded. Experiments with synthetic data. Mostly applied research.

TRL 1: Basic principles observed and reported

Basic properties of algorithms, representations & concepts. Mathematical formulations. Mix of basic and applied research.